Feb 20th

Thursday, February 20, 2014 10:03 AM

Scheduling Algorithms

- FCFS
  - Non-Preemptive
  - If you have large jobs followed by short jobs that don't run is the convoy effect
- SJF
  - Non-Preemptive
- SRTN
  - Preemptive
- Priority Queue
  - Both Preemptive and non-Preemptive
  - Explicit Priority
- Round Robin
  - Preemptive algorithm
  - Time Quantum
    - Units of time, when my process will run an nothing will bother it
      - Only stopped with interrupts
      - Or process termination
      - Ours will have 4ms (milliseconds) to run
  - Almost the preemptive version of first come first server
  - Starvation: when a process doesn't run when it has been waiting
  - Prevents starvation by only letting a process for a certain amount of time
  - Response Time = How much time from submission to start avg = 4.5
  - Turn Around = submission to completion avg = 18
  - The benefit is that there is no starvation
  - The turnaround time for processes late in the queue may suffer
  - the time quantum will need to be adjusted based on the type of system
  - Wait time = time spend not in the run queue (turnaround - burst) avg = 12ms,
  - The convoy affect can happen if we keep increasing the quantum time

Submission	Burst
0	8
1	4
2	9
3	5

Job	Resp	TurnAround	Wait time	
0	0ms	20ms	12	
1	6ms	7ms	3	
2	6ms	24ms	15	
3	922	22ms	17	

- **** we haven't talked about context switching***
  - If we have a long quantum time then we can end up with starvation.
  - The less switches the faster it will go
  - Thus we need to balance the time and switches
  - .
- How to program this

READY QUEUE

Index	Remaining
0	8
1	4
2	9
3	5
0	4

Input
Submission Burst

Submission	Burst
0ms	8ms
1ms	4ms
2ms	9ms
3ms	5ms

TIME Stats

Response	Turnaround time

clock = 0;
Set current job to none

1. Add all jobs to the ready queue
   a. Move current job to the bottom of the queue(if there is a current job)
   b. Select new job to run
      i. If(remaining < = queue)
         1) clock += remaining;
         2) Remove job from ready queue
      ii. Else
         1) clock = clock + timequantum
         2) Remaining -= quantum
         3) Add new jobs to the list

4) Current job = moved to end of ready queue (it's really a circular queue), setting no current job;

c. Repeat the algorithm

2. If you want to change it to shortest job first
    a. Where we move the current job to the end, we bubble the lowest to the top of the queue

Keep track of switch time by adding the context switching element

We will need to do ls - L | wc